


# 1 hour

NAME(Arabic)	TANTA UNIVERSITY FACULTY OF ENGINEERING COMPUTER AND CONTROL DEPARTMENT 3 <sup>RD</sup> GRADE - DATABASE SYSTEMS - QUIZ 1	
Section		

## EmployeeDetail table

EmployeeID	FirstName	LastName	Salary	JoiningDate	Department	Gender
1	Vikas	Ahlawat	600000.00	2013-02-15 11:16:28.290	IT	Male
2	nikita	Jain	530000.00	2014-01-09 17:31:07.793	HR	Female
3	Ashish	Kumar	1000000.00	2014-01-09 10:05:07.793	IT	Male
4	Nikhil	Shama	480000.00	2014-01-09 09:00:07.793	HR	Male
5	anish	kadian	500000.00	2014-01-09 09:31:07.793	Payroll	Male

## ProjectDetail table

ProjectDetailID	EmployeeDetailID	ProjectName
1	1	Task Track
2	2	CLP
3	3	Survey Managment
4	4	HR Managment
5	5	Task Track
6	6	GRS
7	7	DDS
8	8	HR Managment
9	9	GL Managment

- Write down the query to create employee table with Identity column ([EmployeeID])

```
CREATE TABLE EmployeeDetail(
[EmployeeID] INT IDENTITY(1,1) NOT NULL,
[FirstName] NVARCHAR(50) NULL,
[LastName] NVARCHAR(50) NULL,
[Salary] DECIMAL(10, 2) NULL,
[JoiningDate] DATETIME NULL,
[Department] NVARCHAR(20) NULL,
[Gender] VARCHAR(10) NULL
)
```

- How to set foreignkey relationship using query(set EmployeeID column of ProjectDetail table as a foreignkey)

```
ALTER TABLE ProjectDetail
ADD CONSTRAINT fk_EmployeeDetailID_Eid
FOREIGN KEY(EmployeeDetailID)REFERENCES EmployeeDetail(EmployeeID)
```

- Write a query to get only "FirstName" column from "EmployeeDetail" table

```
SELECT FirstName FROM [EmployeeDetail]
```

- Select employee detail whose name is "Vikas" and whose "FirstName" start with latter 'a'.

```
SELECT * FROM [EmployeeDetail] WHERE FirstName = 'Vikas' or firstName like 'a%'
```

- Get all employee details from EmployeeDetail table whose "FirstName" contains 'k'

```
SELECT * FROM [EmployeeDetail] WHERE FirstName like '%k%'
```

- Select all employee detail with First name not "Vikas", "Ashish", and "Nikhil".

```
SELECT * FROM [EmployeeDetail] WHERE FirstName NOT IN('Vikas','Ashish','Nikhil')
```

- Select first name from "EmployeeDetail" table prefixed with "Hello "

```
SELECT 'Hello '+FirstName FROM [EmployeeDetail]
```

- Get employee details from "EmployeeDetail" table whose Salary between 500000 than 600000

```
SELECT * FROM [EmployeeDetail] WHERE salary between 50000 and 60000
```

- Select second highest salary from "EmployeeDetail" table.

```
SELECT TOP 1 Salary
FROM (SELECT TOP 2 Salary FROM [EmployeeDetail] ORDER BY Salary DESC) T
ORDER BY Salary ASC
```

- Get employee name, project name order by firstname from "EmployeeDetail" and "ProjectDetail" for those employee which have assigned project already.

```
SELECT FirstName,ProjectName FROM [EmployeeDetail] A INNER JOIN [ProjectDetail] B
ON A.EmployeeID = B.EmployeeDetailID ORDER BY FirstName
```

11. Get employee name, project name order by firstname from "EmployeeDetail" and "ProjectDetail" for all employee even they have not assigned project.

```
SELECT FirstName,ProjectName FROM [EmployeeDetail] A LEFT OUTER JOIN [ProjectDetail] B
ON A.EmployeeID = B.EmployeeDetailID ORDER BY FirstName
```

12. Get employee name, project name order by firstname from "EmployeeDetail" and "ProjectDetail" for all employee if project is not assigned then display "-No Project Assigned".

```
SELECT FirstName, ISNULL(ProjectName, '-No Project Assigned')
FROM [EmployeeDetail] A LEFT OUTER JOIN [ProjectDetail] B
ON A.EmployeeID = B.EmployeeDetailID ORDER BY FirstName
```

13. Get all project name even they have not matching any employeeid, in left table, order by firstname from "EmployeeDetail" and "ProjectDetail".

```
SELECT FirstName,ProjectName FROM [EmployeeDetail] A RIGHT OUTER JOIN [ProjectDetail] B
ON A.EmployeeID = B.EmployeeDetailID ORDER BY FirstName
```

14. Write a query to find out the employee name who has not assigned any project, and display "-No Project Assigned"(tables :- [EmployeeDetail],[ProjectDetail]).

```
SELECT FirstName, ISNULL(ProjectName, '-No Project Assigned') AS [ProjectName] FROM
[EmployeeDetail] A LEFT OUTER
JOIN [ProjectDetail] B
ON A.EmployeeID = B.EmployeeDetailID
WHERE ProjectName IS NULL
```

15. Write down the query to fetch ProjectName on which more than one employee are working along with EmployeeName.

```
select P.ProjectName, E.FName from ProjectDetails P INNER JOIN EmployeeDetails E
on p.EmployeeID = E.Id where P.ProjectName in(select ProjectName from ProjectDetails group by
ProjectName having COUNT(*)>1)
```

16. Write the query to get the department and department wise total(sum) salary, display it in ascending order according to salary.

```
SELECT Department, SUM(Salary) AS [Total Salary] FROM [EmployeeDetail]
GROUP BY Department ORDER BY SUM(Salary) ASC
```

17. Get department wise maximum salary from "EmployeeDetail" table order by salary ascending

```
SELECT Department, MAX(Salary) AS [Max Salary] FROM [EmployeeDetail]
GROUP BY Department ORDER BY MAX(Salary) ASC
```

18. Write down the query to fetch Project name assign to more than one Employee

```
Select ProjectName,Count(*) [No of Emp] from [ProjectDetail] GROUP BY ProjectName HAVING
COUNT(*)>1
```

19. Write down the query to fetch EmployeeName & Project who has assign more than one project.

```
Select EmployeeID, FirstName, ProjectName from [EmployeeDetail] E INNER JOIN [ProjectDetail] P
ON E.EmployeeID = P.EmployeeDetailID
WHERE EmployeeID IN (SELECT EmployeeDetailID FROM [ProjectDetail] GROUP BY
EmployeeDetailID HAVING COUNT(*) >1 )
```

20. Write the query to get the department, total no. of departments, total(sum) salary with respect to department from "EmployeeDetail" table.

```
SELECT Department, COUNT(*) AS [Dept Counts], SUM(Salary) AS [Total Salary]
FROM [EmployeeDetail] GROUP BY Department
```